A Comparison of Acoustic Models Based on Neural Networks and Gaussian Mixtures

Tomáš Pavelka and Kamil Ekštein

University of West Bohemia, Faculty of Applied Sciences,
Deptartment of Computer Science and Engineering
{tpavelka, kekstein}@kiv.zcu.cz

Abstract. This article tries to compare the performance of neural network and Gaussian mixture acoustic models (GMMs). We have carried out tests which match up various models in terms of speed and achieved recognition accuracy. Since the speed-accuracy trade-off is not only dependent on the acoustic model itself, but also on the settings of decoder parameters, we have suggested a comparison based on equal number of active states during the decoding search. Statistical significance measures are also discussed and a new method for confidence interval computation is introduced.

1 Introduction

The most widely used mathematical framework for automatic speech recognition are the continuous density hidden Markov Models (CDHMMS). Despite of their success these models make various assumptions that are not true for speech data (see [4]). There are attempts to solve some of the drawbacks of the CDHMM paradigm by employing neural networks. The research into the so called *hybrid systems* (see e.g. [1], [8]) has shown that it can be advantageous to use neural networks (instead of the more traditional Gaussian mixtures) as emission probability estimators for hidden Markov model based automatic speech recognizers. Our results presented in [5] demonstrate that there are two main benefits:

- The application of neural networks to emission probability estimation does not place any constraints on the form of its inputs (as opposed to GMM models with diagonal covariance matrices which add delta and acceleration coefficients to the input vector because the elements of the final composed vector are loosely uncorrelated). This is usually exploited by presenting several subsequent speech frames to the input of the neural network and thus allowing the network to "see" a larger context of the speech signal.
- When compared to Gaussian mixture based acoustic models the neural networks need less trainable parameters to achieve similar or better recognition accuracy. As we will show, this can lead to faster recognition speed.

While the above stated can be said about context independent phonetic units (monophones) the experiments presented in [2] make clear that much better

results can be gained with adding context dependency (e.g. by using triphones). In [7] a solution is proposed that attempts to use decision tree clustering for the reduction of the number of physical models in order to solve the sparse data problem and also in order to limit the number of output neurons and thus decrease the time needed for training. In this article we would like to further explore the difficulties in comparing acoustic models based on neural networks and GMMs.

2 Speech Corpora

All the available speech data is in Czech language, recorded in quiet environment at 16 KHz sampling rate and 16 bits per sample. The corpora are divided into sentences; each sentence is stored in a separate file. The training set consists of three parts:

- **Train Schedule Queries**. This corpus consists of questions about train schedules and related information. An example of such question would be "When does the train for Plzeň leave".
- **LAC-HP Chess**. Stands for LASER Audiocorpus High Precision. The corpus was recorded in an audio studio; all the audio files have been verified during the recording. This set consists of voice commands for a chess game. The commands could be either chess moves (e.g. "Move the king to b5") or miscellaneous commands like "I want to start a new game".
- **LAC-HP Phonetic**. This is a set of nonsense sentences with words containing infrequent phonetic units.

The testing corpus for the train schedules is a subset taken out from the original corpus. The testing corpus for the chess game contains only move commands because we have found out that other commands can skew the recognition results (the move commands are much harder to recognize). This means that if other commands are present in the training data the resulting accuracy is highly correlated with moves / other commands ratio. Table 1 shows statistics for all the speech data used in our experiments.

Training Corpus	**Speakers**	**No. of sentences**	**Vocabulary size [words]**	**Total Length [hours]**
Train Schedules	81 (48M, 33F)	11270	1490	11:28:06
LAC-HP Chess	81 (31M, 50F)	2050	96	1:51:50
LAC-HP Phonetic	81 (31M, 50F)	1200	96	1:33:02
Testing Corpus	**Speakers**	**No. of sentences**	**Vocabulary size [words]**	**Total Length [hours]**
Train Schedules	4 (2M, 2F)	400	1490	0:31:34
Chess Moves	20 (10M, 10F)	2000	96	1:18:28

Table 1. Training and testing corpora

3 Gaussian Mixture Acoustic Models

Gaussian mixture models (GMMs) were trained by the Hidden Markov Toolkit (HTK, [9]). Only models with diagonal covariance matrices were tested. The parameter estimation was done by a flat start embedded training which only requires the phonetic transcriptions of the training utterances to be available. On the other hand the neural network needs exact locations of phonetic units in the training data. This leads to the second reason for having a set of trained GMM models: the GMM based recognizer can be used to label the training data for the neural network (by the means of forced Viterbi alignment).

In the case of the GMMs the whole set consists of 36 (35 context independent units + silence) phonetic unit models. Each phonetic unit is a three state HMM, each state has its own mixture of Gaussians. The training starts with one Gaussian per state. In each training cycle embedded re-estimation is performed four times (our tests show that the error decrease after four iterations is negligible). After the cycle is completed the number of Gaussians for each state is increased twofold. We have trained models with up to 128 Gaussians per state.

The training of triphone GMM models is described in detail in [2]. The process is similar to the training of the monophones, the difference is that after the models with single Gaussian per state are trained a decision tree clustering of all the states is performed. The result is that the triphone models which do not have sufficient amount of training data available are tied together with all the other models in their respective clusters. The clustering provides a mapping between the logical models (i.e. any triphone) and the physical models (actual Gaussian mixtures).

The clustering algorithm uses yes-no questions about triphone context which are used to split the models into two parts. A measure exists (see [3]) that evaluates the information gained by the split and thus the best question can be chosen. After that the same is done for each of the newly created model and a tree structure is created. This process is eventually stopped based on a threshold which sets the minimum information gain needed for a split to be allowed. By setting the value of this threshold one can control the total number of physical states (see Table 2).

Threshold	Number of	Best results	
	state models	Train Schedules	Chess Moves
1000	1733	84.46 @ 16mix	97.55 @ 16mix
2000	1070	83.93 @ 32mix	97.51 @ 32mix
5000	517	82.46 @ 32mix	97.52 @ 32mix

Table 2. Accuracy achieved for different values of decision tree clustering threshold.

4 Neural Network Acoustic Models

All networks discussed in this paper are multi layer perceptrons with nine consecutive speech frames on input (altogether 177 neurons). Various numbers of hidden neurons were tested. The number of output vectors corresponds to the number of phonetic units (36 for the monophone case) or the number of physical states (for the triphone case).

Hidden Markov models work with likelihoods $p(\text{input}|\text{class})$ instead of the class posteriors $p(\text{class}|\text{input})$ that we get on the output of the neural network. These can be converted using the Bayes theorem:

$$p(\text{input}|\text{class}) = \frac{p(\text{class}|\text{input}) \cdot p(\text{input})}{p(\text{class})} \quad . \tag{1}$$

Since the probability of an input $P(\text{input})$ is the same for all HMM states examined in a given frame it can be discarded from the equation without affecting the result. Our experiments (details in [7]) have shown that the division by priors is not always beneficial and depends on the testing corpus. In the case of the train schedule corpus the division by priors increases accuracy but in the case of the chess corpus its use is actually harmful. For this reason the division by priors was only done in tests with the train schedule corpus.

The GMMs with the lowest number of physical states (517, see Table 2) were used to generate training vector labels. The resulting triphone neural network had 2000 hidden neurons and 517 output neurons.

5 Confidence Intervals

The problem with the results of speech recognition experiments is that accuracies obtained by tests with various models can be close together and random errors (i.e. caused by the choice of the testing corpus which should be representative of the domain but is always limited in size) should be taken into account. A possible solution would be to treat the recognition results as a random variable and compute confidence intervals for its distribution.

However, the standard binomial distribution confidence interval cannot be used, because it assumes statistical independence of individual errors of which the proportion (i.e. error rate or accuracy) is computed. This is clearly not true in speech recognition where each error can have influence on subsequent errors. We can, however, assume independence of errors in different sentences since all search variables are discarded after a sentence is recognized and there is no way a recognition of one sentence can affect recognition of the next one.

The statistical properties of a sample of independent observations is investigated in [10]: Let $(r_1, s_1), (r_2, s_2), \ldots, (r_n, s_n)$ be the sample where (in the case of speech recognition) r_i is the number of correctly recognized words in sentence i and s_i is the number of incorrectly recognized words. The distribution function

of the variable

$$\xi_n = \frac{\sum_{i=1}^{n} r_i}{\sum_{i=1}^{n} r_i + \sum_{i=1}^{n} s_i} \tag{2}$$

is derived in [10] which can be used to find the boundaries of a confidence interval. If we know the expected value estimations

$$e_r = \frac{1}{n}\sum_{i=1}^{n} r_i, \quad e_s = \frac{1}{n}\sum_{i=1}^{n} s_i \tag{3}$$

as well as the variance estimations

$$\sigma_r^2 = \frac{1}{n}\sum_{i=1}^{n}(r_i - e_r)^2, \quad \sigma_s^2 = \frac{1}{n}\sum_{i=1}^{n}(s_i - e_r)^2 \tag{4}$$

and the correlation coefficient

$$\rho_{r,s}\sigma_r\sigma_s = \frac{1}{n}\sum_{i=1}^{n}(r_i - e_r)(s_i - e_s). \tag{5}$$

then the asymptotic distribution function (for sufficiently large n) of the random variable ξ_n can be expressed (see [10] to see how it is inferred) as

$$F_{\xi_n} = 1 - \Phi\left(-\sqrt{n}\frac{e_s - \left(\frac{1}{x} - 1\right)e_r}{\sqrt{\sigma_s^2 + \left(\frac{1}{x} - 1\right)^2\sigma_r^2 - 2\left(\frac{1}{x} - 1\right)\rho_{r,s}\sigma_r\sigma_s}}\right) \Leftrightarrow 0 < x < 1. \tag{6}$$

We are looking for a confidence interval $(x_L; x_U)$ that satisfies the condition

$$P(x_L \leq \xi_n \leq x_U) = 1 - \alpha. \tag{7}$$

To uniquely determine the bounds we chose α_1 and α_2 so that $\alpha = \alpha_1 + \alpha_2$; $0 < \alpha, \alpha_1, \alpha_2 < 1$. The interval boundaries can then be found by solving the equations

$$F_{\xi_n}(x_U) = 1 - \alpha_1 \tag{8}$$

and

$$F_{\xi_n}(x_L) = \alpha_2. \tag{9}$$

In all our experiments the value of α was set to 0.005 which corresponds to 99% confidence. See [10] for details of the interval computation.

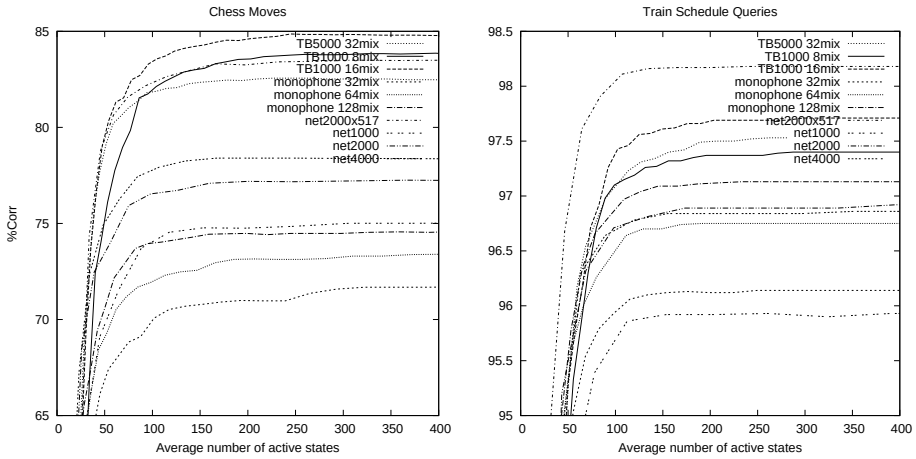

Fig. 1. The relationship between the average number of active states (controlled by the beam threshold) and the resulting accuracy.

6 Experimental Results

Even though the training of the GMM models was done by the HTK software the testing of both the GMM and MLP acoustic models was carried out with the JLASER [6] recognizer. For both acoustic models Mel-frequency cepstral coefficients (MFCCs) served as input.

There was a grammar representing all the possible utterances in the chess moves test corpus but the tests with the train schedule corpus were run without any language model. We do not consider the lack of language model to be a problem since our main goal is to compare the two kinds of acoustic models.

For all tests pruning was performed during the decoding phase. For the train schedule corpus a word insertion penalty was applied. Both the pruning threshold and the word insertion penalty were tuned for each acoustic model in order to achieve the highest possible speed while maintaining the highest recognition accuracy. Decoding with pruning means that the emission probabilities are usually not needed only for all phonetic units (this is especially true for triphones). Some computation can be avoided by computing only those emission probabilities that are requested by the decoding algorithm. This is quite straightforward in the case of GMM models. In the case of neural networks the activations of all the hidden neurons need to be computed for every speech frame. But the computation of the output layer neuron activations can be delayed until those are requested by the decoder.

Another problem with pruning is that the choice of pruning threshold allows one to make trade-off between speed and accuracy which makes the comparison in terms of speed and accuracy more complicated. One thing we have observed is that accuracy rises with less stricter threshold but eventually stabilizes and more loosening of the threshold only decreases speed. So our solution is to compare the

models at this point. But, the stabilizing point is different for different corpora and different models. We have found out that it is best to view the relationship between the average number of active states (as opposed to the actual threshold value) and the resulting accuracy. It can be seen from Figure 1 that this stabilization point is around 200 average active states regardless of corpus and model choice. For this reason, all the models are tested with a beam threshold that leads to an average of 200 active states during decoding.

Figure 2 shows the results for models with different numbers of trainable parameters (controlled by the number of hidden neurons or the number of mixtures and clustering threshold). Besides showing the recognition accuracy the figure also shows the recognition speed measured as a percentage of real time processing power on a referential machine needed for the recognition.

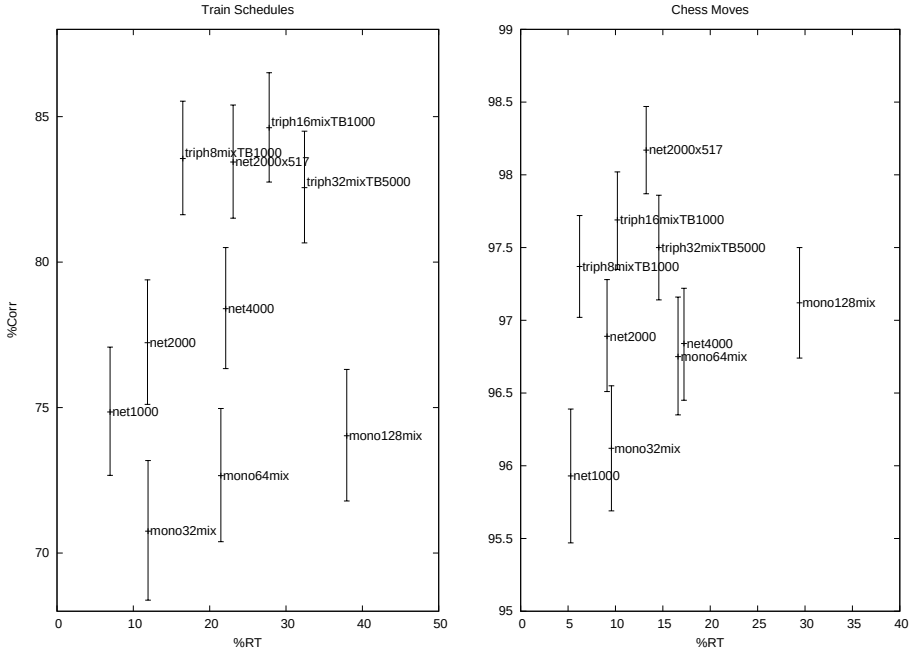

Fig. 2. Comparison of recognition speed and percentage of correct results for all tested acoustic models. The error bars represent the confidence intervals for 99% probability. Neural networks are specified by two numbers: the first denotes the number of hidden neurons and the second the number of output neurons. GMMs are specified by a prefix ("mono" for monophones, "triph" for triphones), the number of mixtures and, in the case of triphones the clustering threshold.

7 Conclusions

The tests reported in [7] were favorable towards neural networks and showed that neural network based acoustic models usually led to higher recognition speeds. In this article we have tested GMM based models with different values for the decision tree clustering threshold which, in some cases (see Figure 2) led to higher recognition speeds. But, there is still an outlying result achieved for the chess corpus where the neural network triphone acoustic model clearly outperforms all other models. This may suggest that triphone neural networks may be suitable for small vocabulary grammar based applications (such as our voice controlled chess game). For LVCSR tasks the Gaussian mixture models provide more flexibility without compromising speed.

We have also suggested an approach to speed vs. accuracy testing based on controlling of the number of active states during the decoding phase. A novel method for confidence interval computation based on findings in [10] has been shown in this article.

Acknowledgement

This work was supported by grant no. 2C06009 Cot-Sewing.

References

1. Bourlard, H. and Morgan, N.: Hybrid HMM/ANN Systems for Speech Recognition: Overview and New Research Directions, Summer School on Neural Networks, 1997.
2. Hejtmánek, J., Pavelka, T.: Use of context-dependent units in Czech speech, Proc. of PhD Workshop 2007, Balatonfüred, Hungary, 2007.
3. Odell, J.J.: *The Use of Context in Large Vocabulary Speech Recognition*, PhD Thesis, Cambridge University Engineering Dept, 1995.
4. Rabiner, L.R.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, *Proceedings of the IEEE*, vol. 77, no. 2, 1989
5. Pavelka, T., Ekštein, K.: Neural Network Acoustic Model for Recognition of Czech Speech, Proc. of PhD Workshop Systems & Control, Izola, Slovenia, 2005.
6. Pavelka, T., Ekštein, K.: JLASER: An Automatic Speech Recognizer Written in Java, Proc. of XII International Conference Speech and Computer (SPECOM'2007), Moscow, Russia, 2007.
7. Pavelka, T., Král, P.: Neural Network Acoustic Model with Decision Tree Clustered Triphones, *Proceedings of 2008 IEEE International Workshop on Machine Learning for Signal Processing*, Cancún, Mexico, 2008.
8. Tebelskis, J.: Speech Recognition using Neural Networks, PhD Thesis, Carnegie Mellon University, 1995.
9. Young, S. et al., The HTK Book (for HTK v. 3.3), Cambridge University Engineering Dept, 2002.
10. Vávra, F., Pavelka, T., Šedivá, B., Vokáčová, K., Marek, P., Neumanová, M.: Ratio Statistics, *Proceedings of JČMF ROBUST 2008*, Pribylina, Slovakia, 2008.